



**CDN live EMEA 2007**

**Functional Verification Track**

**Session 3.8 : HW-SW IP verification flow using ISX**

Laurent DUCOUSSO

# Summary

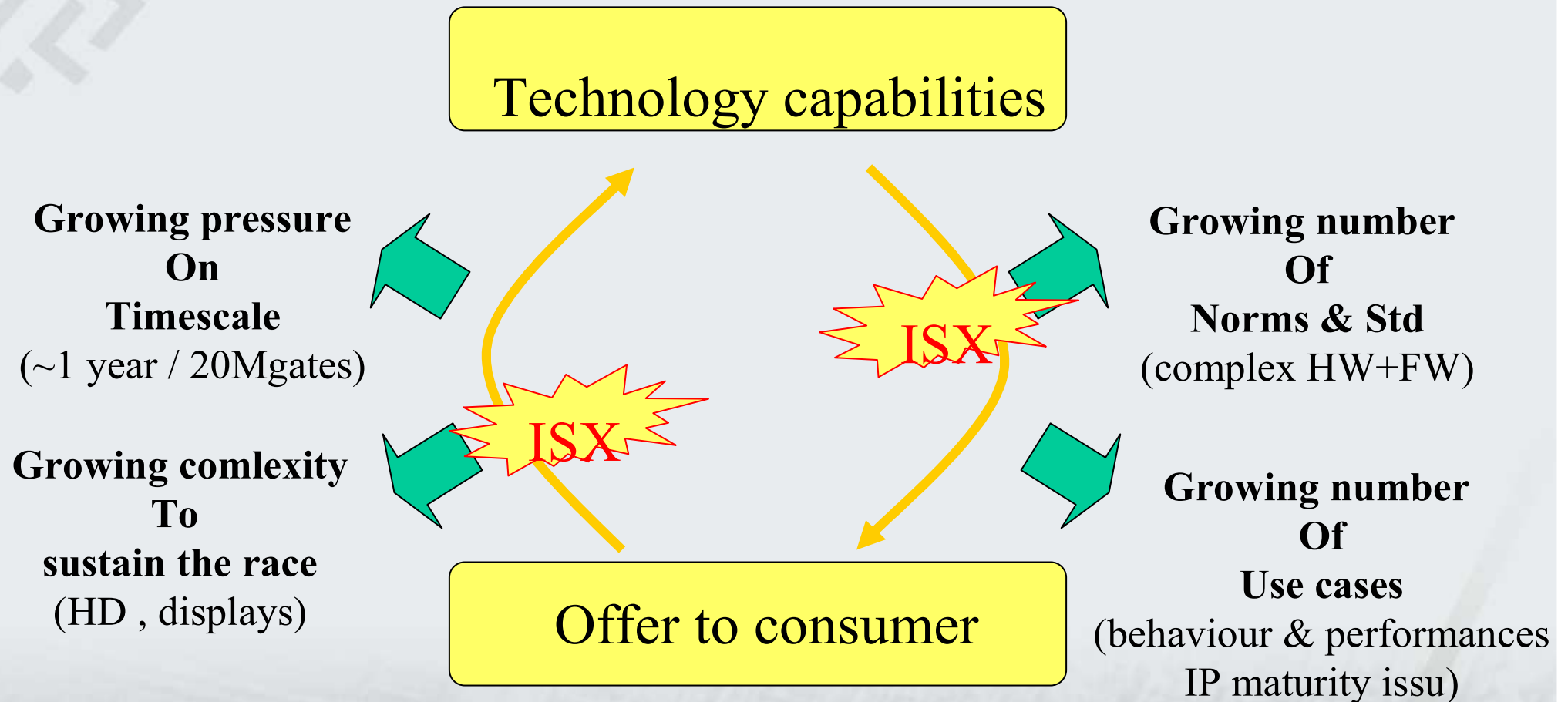
- Thank's to Giles Hall from Cadence (support & slides)
- Products & IP's evolution , need for ISX
- Project & Ip Verification flow : ISX extension
  - Strategy
  - Environment
  - TLM platform
  - SW interfaces
  - Flow
  - Scenario Builder and Test file structure
  - Coverage
- Conclusions

# Home Entertainment Division

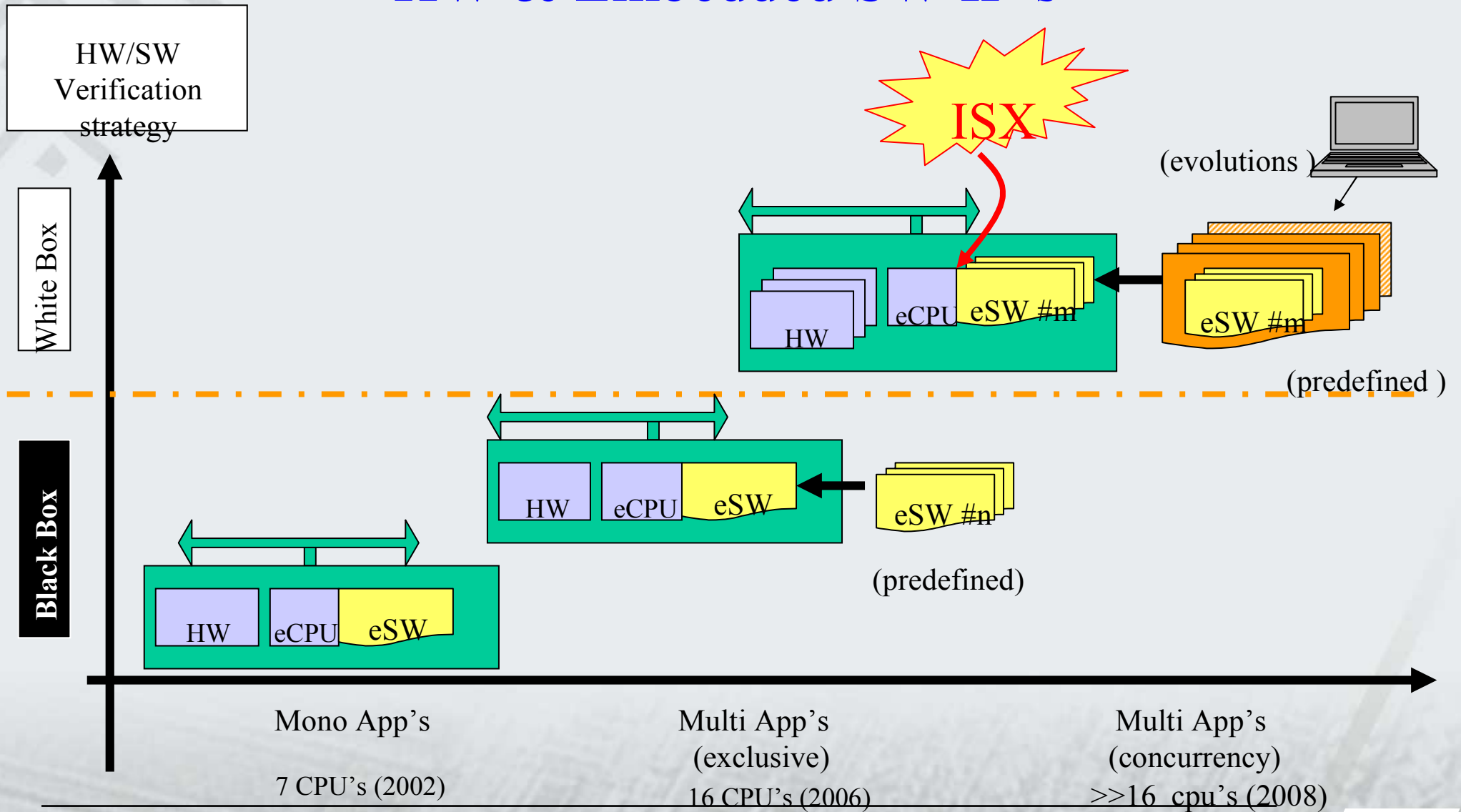


20M gates , 16 Embedded Cpu's

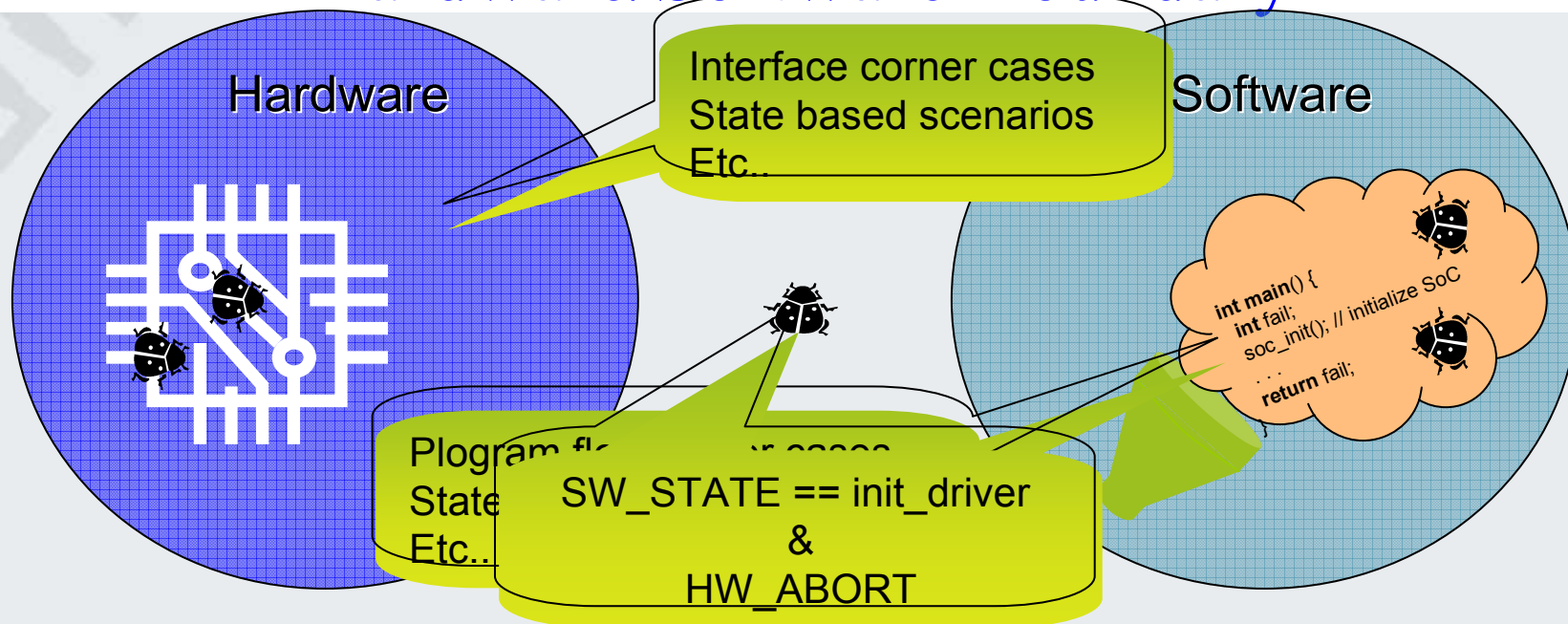
# Verification Challenges



# HW & Embedded SW IP's

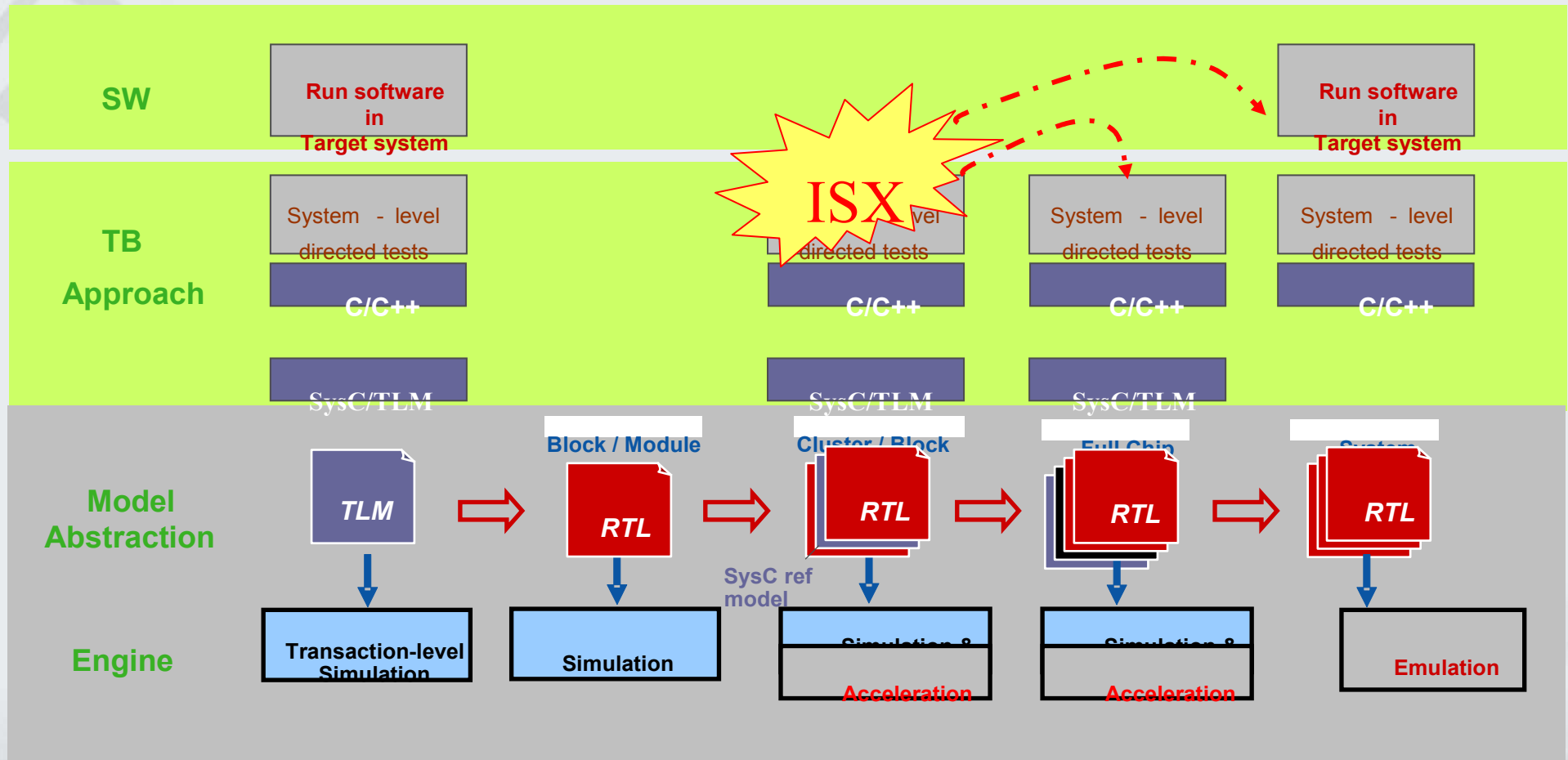


# Hardware/Software Boundary



- How do we stress this third category?
- How do we know we thoroughly stressed it?
- How do we know it behaved correctly?
- **CDV (Coverage Driven Verification) is good at this**
  - How does it work in this heterogeneous world?

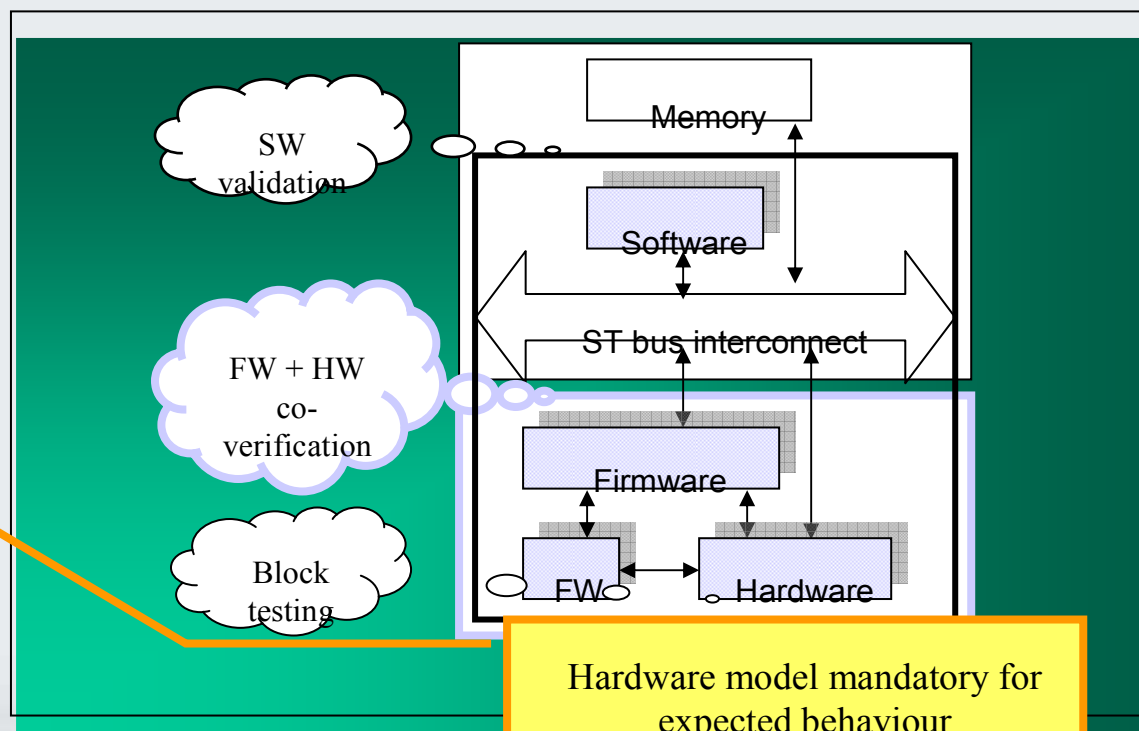
# Project flow



# Verification strategy

- **Verification Strategy**
  - HW block testing
    - Stimuli applied to design
  - FW + HW coverivation
    - App's or algorithm checking
  - SW validation
    - Visual qualification of streams
- **top-down verification**
  - only top level reference model exists
  - maintenance and devt of module env + ref model too costly
- **hw/sw part constantly moving**
  - specs changing
  - requirements changing
- **need to use simulation + emulation/acceleration to get through verification**

HW-FW interface  
Controlability / observability



Hardware model mandatory for expected behaviour

Constraint on Architecture maturity

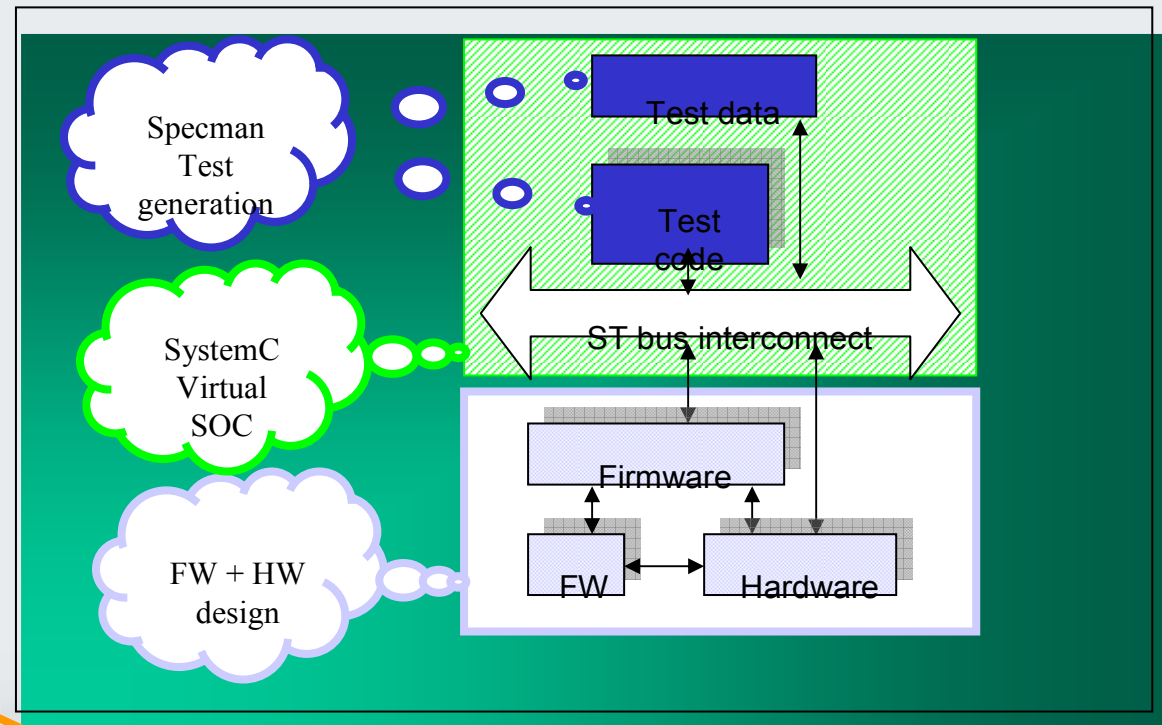
# Verification environment

- **Verification Environment**
  - RTL cores + hardware
  - SystemC virtual SOC
    - On the fly timing randomisation
  - Specman tests generation
    - **Test driven Methodology**
    - Pre-compiled tests
    - C verification driver
    - Random datas
    - + e langage capabilities

ISX

## ISX tests generation

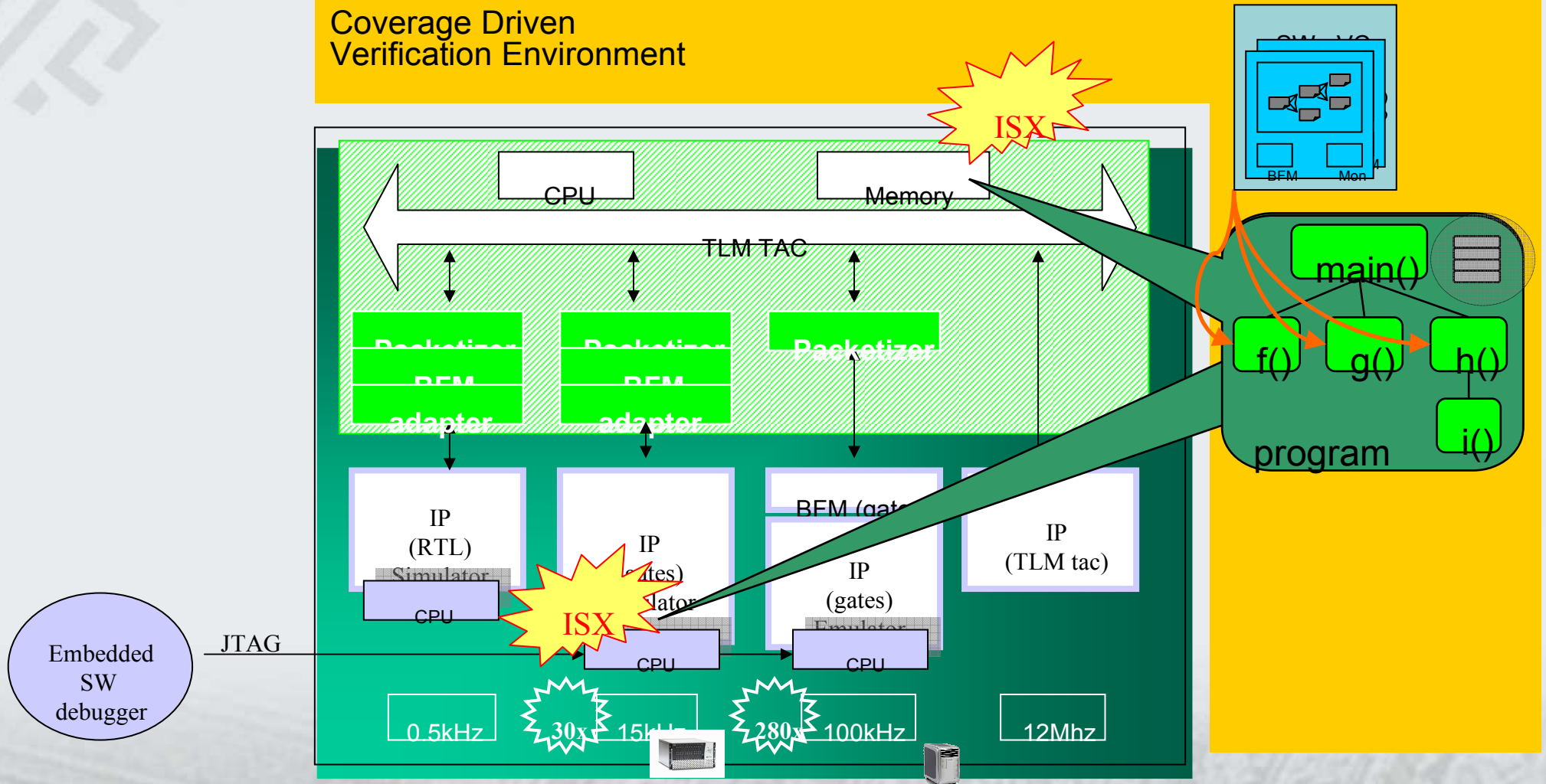
- CDV
- Scenario builder\*
- Random Sequencing
- HW-SW debug



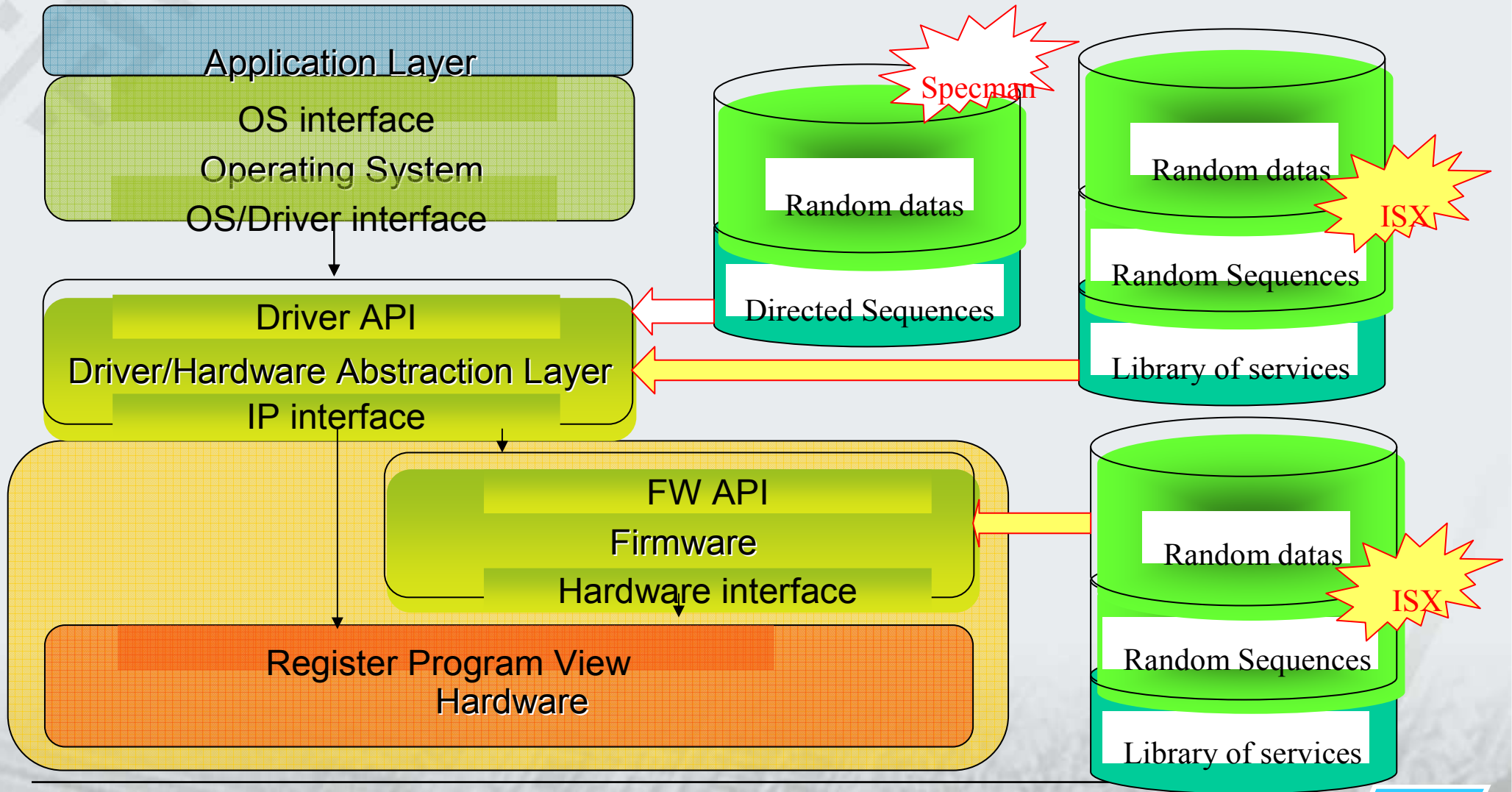
Mixed flow Test Driven + CDV

# TLM Flow & ISX

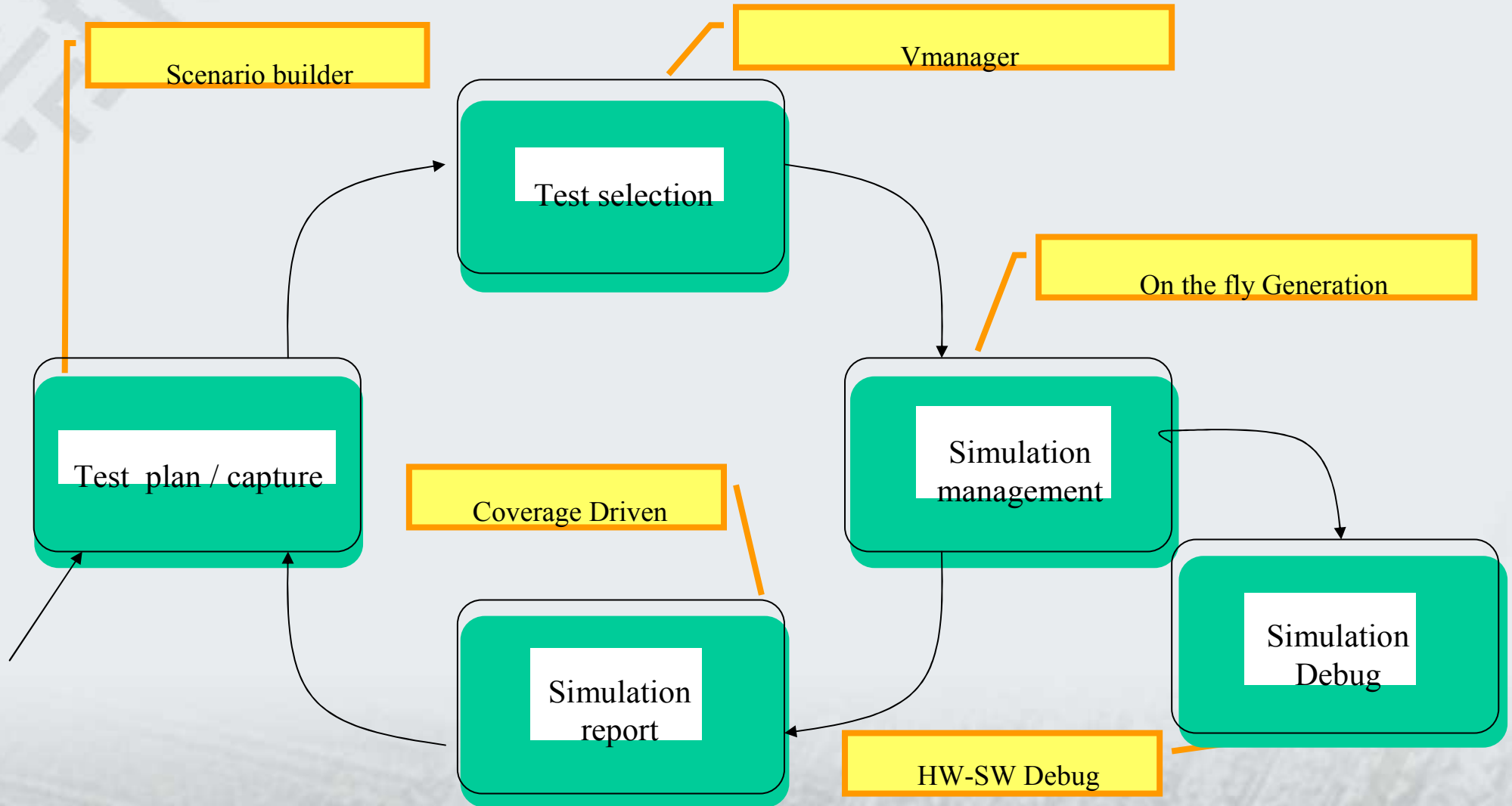
Coverage Driven  
Verification Environment



# Software Verification Interfaces



# Test driven flow -> CDV



# Test scenario

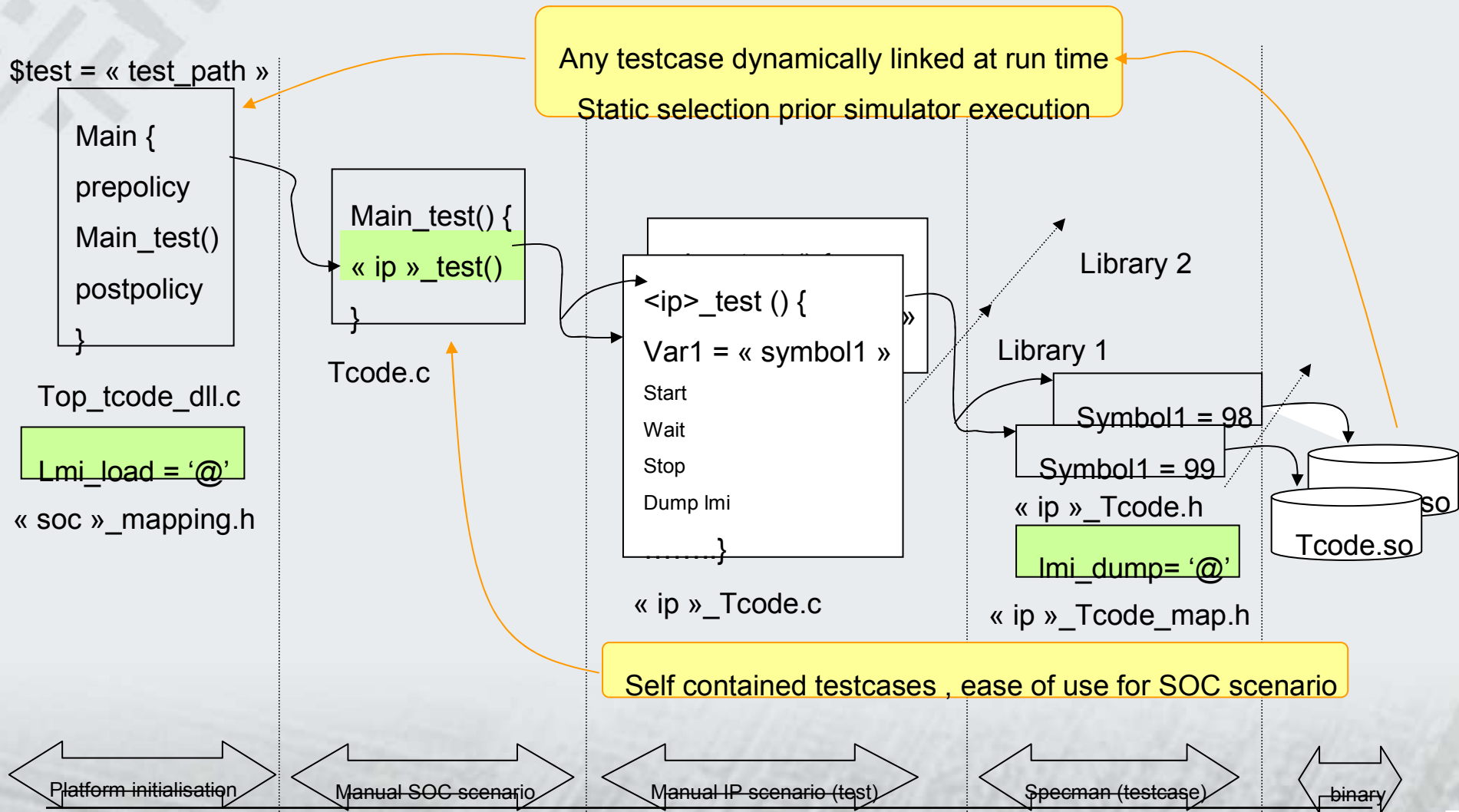
The screenshot displays the Incisive Enterprise Scenario Builder interface. On the left, there are three panels: 'Interface Drivers' with items like 'reg\_driver', 'sw\_drv', and 'virtual\_seq\_driver'; 'Components' with 'Stimulus Items' including 'INIT\_HW', 'PRODUCE\_BILL', 'READ\_PURCHASE', 'READ\_STR', 'UPDATE\_COST\_TABLE', and 'WRITE\_STR'; and 'Stimulus Sequences' with items like 'CLEAR\_PURCHASES', 'INIT\_HW', 'PRODUCE\_BILL', 'READ\_PURCHASE', 'READ\_STR', 'UPDATE\_COST\_TABLE', and 'WRITE\_STR'. The main workspace shows a 'Sequence' editor for 'FRED' with a tree structure containing 'Parallel (All of)' blocks, 'Thread' blocks, and various 'sw\_sequence' items. A 'Constraints for: UPDATE\_COST\_TABLE sw\_sequence' panel is visible at the bottom, showing a 'clist' block with a 'size' constraint of [0..4294967295] and a 'value < 100' constraint. Three yellow callout boxes provide additional context: one points to the 'Interface Drivers' list, another points to the 'Sequence' editor, and a third points to the 'Constraints' panel.

Software scenarios listed with all other scenarios

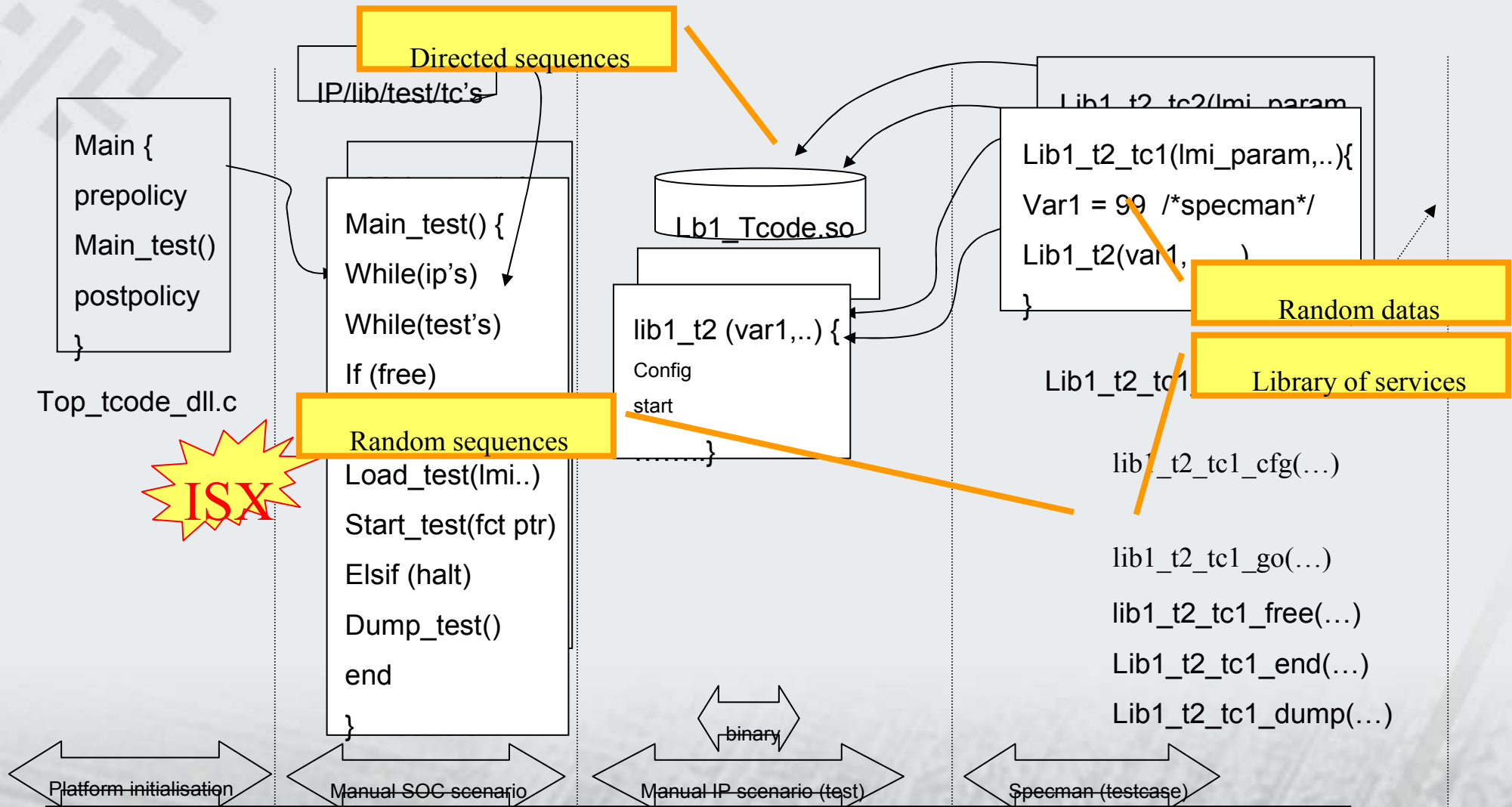
Building reusable sequences or test sequences fully GUI based

Apply constraints with no knowledge of 'e' required

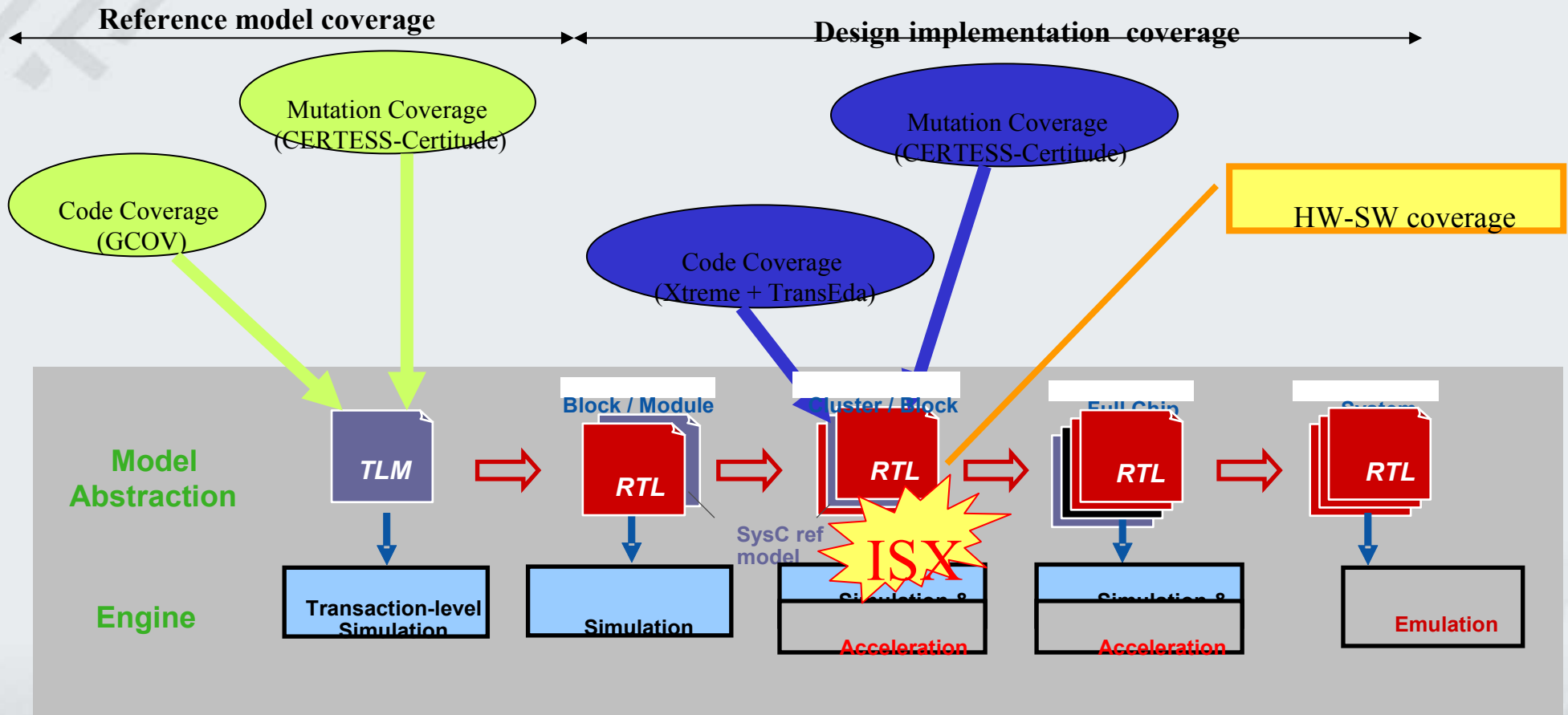
# Directed Sequence : Test code & File Structure



# Random Sequences : Test code & file Structure



# Verification Coverage



# Conclusions

- No Escape to HW/SW interface verification
- New step to the existing flow , new tools
- Reuse of CDV well proven methodology
- Flow in Place on a Pilot Project
- Futur work
  - Waiting pilot project maturity :
    - Xtreme coemulation platform
    - ISX tests Debug phase
    - Big picture on a complete project
  - Maintainability of « ISX » tests

# Thank You !

